Learn Generic Multi-Index Models Near Information-Theoretic Limit

Bohan Zhang[1*], Zihao Wang[2*], Hengyu Fu[3], Jason D. Lee[3]

[1]School of Mathematical Science, Peking University

[2]Department of Mathematics, Stanford University

[3]Electrical Engineering and Computer Sciences, UC Berkeley

Background

Feature Learning

- The success of deep learning is often attributed to the ability of deep neural networks to learn salient *features* of the input.
- Feature learning enables improved accuracy and transfer learning.
- This can lead to sample complexity improvements over kernel methods, which cannot learn features [1].

Learning Linear Features with Two-Layer Networks

- Prior works [2, 3, 4, 5] show that gradient descent on two-layer networks efficiently learns *multi-index models*: $f^*(x) = g^*(Ux)$ where $U \in \mathbb{R}^{r\times d}$ is a low-dimensional projector.
- Gradient descent first learns *linear*, low-dimensional features Ux.
- Once features are learnt, sample complexity to learn g^* is d-independent.

Information-Theoretic Limit

- The information-theoretic limit for learning multi-index models is $\Theta(d)$ samples.
- There exist polynomial time algorithms [6] that achieve this for generic generative-exponent-two multi-index models which covers almost all canonical examples.
- The earliest work [3] showed that two-layer neural networks can efficiently learn multi-index models with a sub-optimal sample complexity $\widetilde{\Theta}(d^2)$.

Key Questions:

Can neural networks trained via standard optimizers (e.g., gradient descent) achieve *efficient, near–information-theoretic-optimal* learning of generic multi-index models?

Setup

Target Functions:

- $f^*(x) = g^*(Ux)$ where $g : \mathbb{R}^p \to \mathbb{R}$. Assume g^* is a degree p polynomial.
- Hidden subspace $U \in \mathbb{R}^{r\times d}$ (orthogonal unit rows).
- r, p treated as constants.

Two-Layer Neural Network:

$$f_\Theta(x) = \sum_{j=1}^{m} a_j \sigma(w_j^T x + b_j)$$

- $a, b \in \mathbb{R}^m, W \in \mathbb{R}^{m\times d}$.

Empirical and population loss

$$\mathcal{L}(\Theta)\colon \mathbb{E}_x[\ell(f_\Theta(x), f^*(x)], \quad \widehat{\mathcal{L}}(\Theta)\colon \frac{1}{n}\sum_{i=1}^{n} \ell(f_\Theta(x_i), f^*(x_i))$$

Algorithm 1: Layer-wise GD on empirical square loss

Symmetry Initialization: a_j Rademacher; w_j uniform on sphere of radius ε_0.

Stage 1: Train W on D_1 with T_1 gradient steps and small learning rate η_1.

Stage 2: Train a on D_2 with T_2 gradient steps and small learning rate η_2.

Such a layer-wise training procedure is common in prior works [2, 3].

Assumptions and Main Theorem

Input Domain: We assume a standard Gaussian input $N(0, I_d)$.

Assumptions:

- **A1 (Link function).** $\mathbb{E}[g^{*2}(z)] \le C$.
- **A2 (Activation).** $\sigma \in C^3, \sigma'(0) = 0, \sigma''(0) = 1,$ and $|\sigma^{(3)}(\cdot)| \le M$.
- **A3 (Loss)** $\ell \in C^2(\mathbb{R}^2)$, $\ell(t, y) = 0$ at $t = y$, convex in the first argument, and $|\partial_t \ell(t, y)|, |\partial_t^2 \ell(t, y)| \le L$. (Alternatively use Assumption 3' allowing $|\partial_t \ell(t, y)| \le L(1 + |t| + |y|)$ with $\beta = 0$.)
- **A4-A5 (Non-degenerate covariance)** Let $\ell_i = \ell'(0, y_i)$, define

$$\widehat{\sum_\ell} = \frac{1}{n}\sum_{i=1}^{n} \ell_i\, x_i x_i^T, \quad \sum_\ell = \mathbb{E}[\ell'(0, y) x x^T].$$

Assume $\mathbb{E}[\ell_i] = 0$ and $\operatorname{rank}(\sum_\ell) = r$. Denote nonzero eigenvalues $1 = \lambda_1 \ge \cdots \ge \lambda_r > 0$, and condition numbers $\kappa = \frac{\lambda_1}{\lambda_r}$.

- **A6 (Monomial approximation property).** The activation admits an expectation-form approximation of monomials z^k with exponent $\beta \ge 0$.
- **A7 (Small initialization).** Initialization level ε_0 is sufficiently small (only required to be polynomially small provided $T_1 = o(\log d)$).

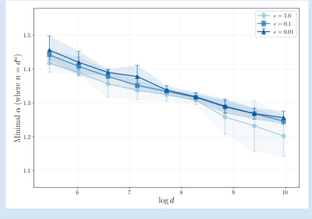
Theorem 1 (informal): Under the above assumptions, for any first-stage time $T_1 = o(\log d)$ and sample size

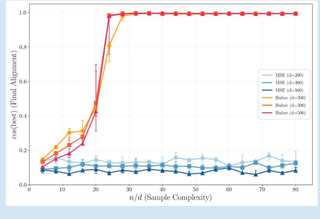
$$n \gtrsim d \log d \cdot T_1^2\, \kappa^{2T_1} + d^{1+1/T_1}\kappa^2,$$

there exists a proper choice of the hyperparameters and second stage training time T_2, with high probability, Algorithm 1 returns

$$\mathcal{L}(\Theta^{(T)}) \lesssim \kappa^{4PT_1}(\log d)^{4p+1}\left(\frac{1}{\sqrt{m}} + \frac{1}{\sqrt{n}}\right)^{\beta+1},$$

Experiments





Exp1: Near-optimal sample complexity

- **Setup:** Target $f^*(x) = \sqrt{5/2}(x_1^2 + \frac{1}{2}x_2^2)$ $(r = 2)$; 2-layer N-N, $m = 4, \sigma(t) = t^2$.
- **Protocol:** Use $n = \lfloor d^\alpha \rfloor$, sweep α, report minimal α to reach test error $\le \alpha$.
- **Findings:** Minimal α decreases with d and trends with $\alpha \approx 1 \Longrightarrow$ support $\widetilde{\Theta}(d^2)$ samples.

Exp2: choose of loss functions matters

- **Setup:** Target $h_4(x_1) + h_4(x_2)$. $m = 4, \sigma(t) = \cos(t)$. Compare Huber vs MSE.
- **Findings**: MSE fails. Huber shows sharp recovery around $n/d \approx 20$.
- **Loss matters:** The loss function can be interpreted as a form of data preprocessing; By properly choosing loss function, we expect $\ell'(0, y)$ exhibits pronon-degenerate second-order information.

Intuition and Proof Sketch

Stage 1: GD ≈ power iteration.

- With symmetric, small initialization, the network output stays near 0 for a nontrivial horizon, so the dominant term in the first-layer update is linear:
  $W^{(t+1)} \approx \widehat{\sum_\ell} W^{(t)}$ (up to controlled higher-order terms)
- Thus the inner weights can perform a **power-iteration process**. This process implicitly mimics a spectral start for the whole span of the hidden subspace and eventually eliminates finitesample noise and recovers this span.

Optimal training time:

- Training the first layer for too long causes features to align only with dominant directions.
- Training for too few steps leaves noisy directions uneliminated.
- The optimal strategy is to stop at an intermediate time, ensuring the features span the entire subspace while eliminating noise.
- Balancing these yields a T that grows with d (typically $T \approx \Theta(\sqrt{\log d})$)

Stage 2: After Stage 1, we freeze the learned first-layer weights $W = W^{(T_1)}$ and the reinitialized biases, only optimize the outer layer a.
We run GD on the ridge-regularized empirical objective which is convex in a.

$$F(a) = \hat{L}(a, W, b) + \frac{\beta_2}{2}\|a\|^2$$

Key decomposition

Introduce an "ideal" comparator $\Theta^* = (a^*, W, b)$. Then

$$L(\Theta^{(T)}) = \hat{L}(\Theta^*) + (\hat{L}(\Theta^{(T)}) - \hat{L}(\Theta^*)) + (L(\Theta^{(T)}) - \hat{L}(\Theta^{(T)}))$$

- (I) **Representation**: constructing Θ^* : Using the paper's monomial approximation property (Assumption 6) and the bias reinitialization, each monomial can be written (approximately) as an expectation of $\sigma(\cdot)$. Approximating this expectation by m sampled features yields small $\hat{L}(\Theta^*)$.
- (II) **Optimization**: GD finds a near-ERM solution.
- (III) **Generalization**: uniform bound via Rademacher complexity.

Future directions

Why this is interesting. The result shows standard gradient-based training can learn generic multi-index structure near the information-theoretic sample limit: early feature learning behaves like a power method that recovers the full hidden subspace, after which prediction reduces to a (nearly) dimension-free second-stage fit.

Future Directions:

- Extend beyond Gaussian inputs and beyond polynomial links g.
- Remove layer-wise + sample-splitting assumptions and analyze end-to-end training.

References

[1] Greg Yang and Edward J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. ICML, 2021.
[2] Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz. The merged-staircase property: a necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks. COLT, 2022.
[3] Alex Damian, Jason Lee, and Mahdi Soltanolkotabi. Neural networks can learn representations with gradient descent. COLT, 2022.
[4] Jimmy Ba, Murat A. Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. NeurIPS, 2022.
[5] Alberto Bietti, Joan Bruna, Clayton Sanford, and Min Jae Song. Learning single-index models with shallow neural networks. NeurIPS, 2022.
[6]Alex Damian, Jason D. Lee, and Joan Bruna. The generative leap: Sharp sample complexity for efficiently learning gaussian multi-index models, 2025.